IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPEAL BRIEF FOR THE APPELLANT

Ex parte BOSWORTH, *et al.*

CELL BASED DATA PROCESSING

Serial No. 09/741,219
Confirmation No. 7676
Appeal No.:
Group Art Unit: 2193

Please charge Counsel's credit card in the amount of Five Hundred Forty Dollars ($540.00) to cover the official fee for this Appeal Brief. In the event that there may be any fees due with respect to the filing of this paper, please charge Deposit Account No. 50-2222.

/Keith M. Mullervy/
Keith M. Mullervy
Attorney for Appellant(s)
Reg. No. 62,382

SQUIRE, SANDERS & DEMPSEY LLP
8000 Towers Crescent Drive, 14th Floor
Vienna, VA 22182-6212

Atty. Docket: 104402.00119

KMM/jf

Encls: Appeal Brief

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re the Appellant:

Adam BOSWORTH et al. Appeal No.:

Serial Number: 09/741,219 Group Art Unit: 2193

Filed: December 19, 2000 Examiner: Tuan A. Vu

Confirmation No. 7676

For: CELL BASED DATA PROCESSING

BRIEF ON APPEAL

August 27, 2010

This is an appeal from the final rejection set forth in an Official Action dated July 13, 2010 ("Final Office Action"), finally rejecting claims 1, 5-11, and 15-21, all of the claims pending in this application, as being unpatentable over W3C, "XML Path Language (XPath)" and "XSL Transformation (XSLT)", version 1.0; W3C Recommendation 16 November 1999 (collectively, "W3C"). This Appeal Brief is being timely filed.

I. REAL PARTY IN INTEREST

The real party in interest in this application is Oracle International Corporation which owns the assignee of record, BEA Systems, Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no known related appeals and/or interferences which will directly effect or be directly effected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1, 5-11, and 15-21, all of the claims pending in the present application, are the subject of this appeal. Claims 1, 5-11, and 15-21 were rejected under 35 U.S.C. § 103(a) as being unpatentable over W3C. Claims 2-4, 12-14, and 22-25 were previously cancelled by Appellants.

IV. STATUS OF AMENDMENTS

All of claims 1, 5-11, and 15-21 stand as they were previously presented prior to the Final Office Action. No amendments were made after the final rejection. Thus, claims 1, 5-11, and 15-21 are pending, and the rejections of claims 1, 5-11, and 15-21 are appealed.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1, upon which claims 5-10 are dependent, recites a computer-implemented method of cell-based data processing that facilitates the execution of computer programming code by a computer system. *See e.g.,* Specification at page 2, lines 17-20; page 14, lines 3-5; page 15, lines 6-8; and page 16, lines 2-19. The method includes receiving as input computer code a data processing specification comprising a plurality of cells. *See e.g.,* Specification at page 2, lines 17-19; page 5, lines 5-8; and page 14, line

9. Each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells. *See e.g.,* Specification at page 5, lines 8-11. The formula of a first cell may reference a value of a second cell. *See e.g.,* Specification at page 5, lines 11-12. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. *See e.g.,* Specification at page 6, lines 8-9 and 11-12. The method further includes parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow. *See e.g.,* Specification at page 14, line 9 – page 15, line 2. The method further includes executing the computer code of the specification in accordance with the execution flow. *See e.g.,* Specification at page 15, lines 5-8. The executing comprises evaluating the formula of each cell in the execution flow and generating an output result. *See e.g.,* Specification at page 6, lines 11-12; and page 15, lines 5-24. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. *See e.g.,* Specification at page 6, lines 5-7.

Claim 11, upon which claims 15-20 are dependent, recites an apparatus. *See e.g.,* Specification at page 16, lines 1-2. The apparatus includes at least one storage unit having stored thereon programming instructions that are configured to be executed by a computer processor and designed to receive as input computer code a data processing specification comprising a plurality of cells, s*ee e.g.,* Specification at page 2, lines 17-19; page 5, lines 5-8; page 14, line 9; and page 16, lines 4-14, where each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, s*ee e.g.,* Specification at page 5, lines 8-11,

where the formula of a first cell may reference a value of a second cell, s*ee e.g.,* Specification at page 5, lines 11-12. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. *See e.g.,* Specification at page 6, lines 8-9 and 11-12. The programming instructions are further designed to parse the specification to determine an interdependency of the plurality of cells and generate and store a directed graph of the interdependency as an execution flow. *See e.g.,* Specification at page 14, line 9 – page 15, line 2. The programming instructions are further designed to execute the computer code of the specification in accordance with the execution flow, s*ee e.g.,* Specification at page 15, lines 5-8, where the executing comprises evaluating the formula of each cell in the execution flow and generating an output result, s*ee e.g.,* Specification at page 6, lines 11-12; and page 15, lines 5-24. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. *See e.g.,* Specification at page 6, lines 5-7. The apparatus further includes at least one processor coupled to said at least one storage unit to execute said programming instructions. *See e.g.,* Specification at page 16, lines 4 and 8-10.

Claim 21 recites a computer with a memory having stored thereon instructions that when executed cause to the computer to implement data processing. *See e.g.,* Specification at page 16, lines 2-14. The computer includes means for receiving a data processing specification comprising a plurality of cells. *See e.g.,* Specification at page 2, lines 17-19; page 5, lines 5-8; page 14, line 9; and page 16, lines 2-14. Each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells. *See e.g.,* Specification at

page 5, lines 8-11. The formula of a first cell may reference a value of a second cell. *See e.g.,* Specification at page 5, lines 11-12. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. *See e.g.,* Specification at page 6, lines 8-9 and 11-12. The computer further includes means for parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow. *See e.g.,* Specification at page 14, line 9 – page 15, line 2; and page 16, lines 2-14. The computer further includes means for executing the specification in accordance with the execution flow. *See e.g.,* Specification at page 15, lines 5-8; and pages 16, lines 2-4. The executing comprises evaluating the formula of each cell in the execution flow and generating an output result. *See e.g.,* Specification at page 6, lines 11-12; and page 15, lines 5-24. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. *See e.g.,* Specification at page 6, lines 5-7.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The ground of rejection to be reviewed on appeal is the rejection of claims 1, 5-11, and 15-21 under 35 U.S.C. §103(a) as allegedly being unpatentable over W3C. As will be discussed below, this rejection is in error, and claims 1, 5-11, and 15-21 should all be found to meet the U.S. requirements for patentability under 35 U.S.C. § 103.

VII. ARGUMENT

Appellants respectfully submit that each of the rejected claims 1, 5-11, and 15-21 recites patentable subject matter that is not taught, disclosed, or suggested by the cited art. Each of the claims is being argued separately, and thus, each of the claims stands or falls alone.

As reiterated by the Supreme Court in *KSR International Co. v. Teleflex Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007), the framework for an objective analysis for determining obviousness under 35 U.S.C. § 103 is stated in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966). Obviousness is a question of law based on underlying factual inquiries. The factual inquiries are: (a) determining the scope and content of the prior art; (b) ascertaining the differences between the claimed invention and the prior art; and (c) resolving the level of ordinary skill in the pertinent art. *See KSR International Co. v. Teleflex Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007); *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966); see also MPEP § 2141.

The Supreme Court in *KSR* also noted that the analysis supporting a rejection under 35 U.S.C. § 103 should be made explicit. The court stated that "rejections on obviousness cannot be sustained by mere conclusory statements; instead there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *See KSR*, 550 U.S. at 398, 82 UPSQ2d at 1396; see also MPEP § 2141.

A. Claims 1, 5-11, and 15-21 are not obvious in view of W3C

As will be argued below, with respect to each separate claim, W3C fails to disclose

or suggest all of the elements of claims 1, 5-11, and 15-21. Thus, the rejection of the Final Office Action is in error, and claims 1, 5-11, and 15-21 should all be found to meet the U.S. requirements for patentability under 35 U.S.C. § 103.

i) Claim 1

Claim 1 recites a computer-implemented method of cell-based data processing that facilitates the execution of computer programming code by a computer system. The method includes receiving as input computer code a data processing specification comprising a plurality of cells. Each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells. The formula of a first cell may reference a value of a second cell. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. The method further includes parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow. The method further includes executing the computer code of the specification in accordance with the execution flow. The executing comprises evaluating the formula of each cell in the execution flow and generating an output result. Each cell is interlocked with at least one other cell through the formula or attribute of each cell.

W3C refers to a series of documents detailing World Wide Web standards promulgated by the World Wide Web Consortium. In particular, W3C includes two specifications, "XSL Transformation (XSLT)" and "XML Path Language (XPath)." XSLT is a specification which describes a syntax and semantics of XSL Transformations ("XSLT"),

which is a language for transforming extensible Markup Language ("XML") documents into other XML documents. *See* W3C – XSLT at Abstract. XPath is a specification that describes XPath, which is a language for addressing parts of an XML document. *See* W3C – XPath at Abstract.

W3C describes that a transformation expressed in XSLT describes rules for transforming a source tree (e.g., an input XML document) into a result tree (e.g., an output XML document). A transformation is achieved in XSLT by associating patterns with templates, where a pattern is matched against elements in the source tree, and a template is instantiated to create part of the result tree. A transformation is expressed in XSLT as a stylesheet, and a stylesheet contains a set of template rules. A template rule has two parts: a pattern which is matched against nodes in the source tree and a template which is instantiated to form part of the result tree. A template contains elements that specify literal result element structure. A template also contains elements from the XSLT namespace that are instructions for creating result tree fragments. When a template is instantiated, each instruction is executed and replaced by the result tree fragment that it creates. Instructions can select and process descendant source elements. Processing a descendant element creates a result tree fragment by finding the applicable template rule and instantiating its template. Elements are only processed when they have been selected by the execution of an instruction, and the result tree is constructed by finding the template rule for the root node and instantiating its template. *See* W3C – XSLT at Section 1. Introduction.

W3C further describes that XPath provides a common syntax and semantics for functionality shared between XSLT and XPointer. XPath provides basic facilities for

manipulation of strings, numbers and Booleans. XPath models an XML document as a tree of nodes, such as element nodes, attributes nodes and text nodes. XPath defines a way to compute a string-value for each type of node. The primary syntactic construct in XPath is an expression. An expression is evaluated to yield an object, which has one of the following four basic types: node-set (an unordered collection of nodes without duplicates), boolean (true or false), number (a floating-point number), and string (a sequence of characters). Expression evaluation occurs with respect to a context. A context consists of: a node (identified as a context node), a pair of non-zero positive integers (identified as a context position and context size), a set of variable bindings, a function library, and a set of namespace declarations in scope for an expression. *See* W3C – Xpath at Section 1. Introduction.

Appellants respectfully submit that W3C fails to disclose or suggest, "a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell, wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing … wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in claim 1.

As described above, W3C describes an XSLT processing model which includes a source tree (e.g., an input XML document), a spreadsheet (which includes a set of template rules, where each template rule includes a pattern and a template), and a result tree (e.g., an output XML document), where an XLST processor processes a source tree

using a spreadsheet to produce a result tree. The Final Office Action broadly cites various portions of W3C as allegedly disclosing the aforementioned limitations of independent claim 1. More specifically, the Final Office Action alleges that the various components of the XSLT processing model (i.e., source tree, result tree, and style sheet) are analogous to the "data processing specification comprising a plurality of cells" recited in independent claim 1. As will be described below in more detail, this position is erroneous as neither the source tree, result tree, nor stylesheet, include a plurality of cells:

- "wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell,"
- "wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing"; and
- "wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in independent claim 1.

W3C merely describes a source tree as an object that includes nodes, where each node includes data elements, where elements from the source tree can be filtered and reordered by a style sheet, and arbitrary structure can be added. *See* W3C – XSLT at Section 1. Introduction. W3C further provides an example of a source tree, where a source tree is a simple XML document (*see* W3C at Section D.2. Data Example):

```
<sales>

    <division id="North">
        <revenue>10</revenue>
        <growth>9</growth>
        <bonus>7</bonus>
    </division>

    <division id="South">
        <revenue>4</revenue>
        <growth>3</growth>
        <bonus>4</bonus>
    </division>

    <division id="West">
        <revenue>6</revenue>
        <growth>-1.5</growth>
        <bonus>2</bonus>
    </division>

</sales>
```

As can be seen from the example provided in W3C, the source tree merely includes nodes, where each node include data elements. The source tree does not include a formula specifying an action or computation to perform when the cell is executed, and does not include one or more attributes referencing other cells. The source tree further does not include a cell reserved as an output cell for outputting a result of processing. Finally, the source tree does not include a plurality of cells, where each cell is interlocked with at least one other cell through the formula or attribute of each cell.

Thus, the source tree described in W3C does not disclose or suggest "a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell, wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing … wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in claim 1.

Furthermore, W3C merely describes a result tree as an object that includes fragments, where each fragment is a result of an instantiated template of a stylesheet applied to a data element of a source tree. *See* W3C – XSLT at Section 1. Introduction. W3C further provides an example of a result tree, where a result tree is a simple HTML document (*see* W3C at Section D.2 Data Example).

```
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Sales Results By Division</title>
</head>
<body>
<table border="1">
<tr>
<th>Division</th><th>Revenue</th><th>Growth</th><th>Bonus</th>
</tr>
<tr>
<td><em>North</em></td><td>10</td><td>9</td><td>7</td>
</tr>
<tr>
<td><em>West</em></td><td>6</td><td style="color:red">-1.5</td><td>2</td>
</tr>
<tr>
<td><em>South</em></td><td>4</td><td>3</td><td>4</td>
</tr>
```

</table>
</body>
</html>

As can be seen from the example provided in W3C, the result tree merely includes fragments, where each fragment is created by applying a template of a stylesheet to a data element of the source tree. The result tree does not include a formula specifying an action or computation to perform when the cell is executed, and does not include one or more attributes referencing other cells. The result tree further does not include a cell reserved as an output cell for outputting a result of processing. Finally, the result tree does not include a plurality of cells, where each cell is interlocked with at least one other cell through the formula or attribute of each cell. Thus, the result tree described in W3C also does not disclose or suggest the aforementioned limitations of claim 1.

With respect to a stylesheet, W3C describes a stylesheet as containing a set of template rules. A template rule has two parts: a pattern which is matched against nodes in the source tree and a template which is instantiated to form a fragment of a result tree. A template can contain elements that specify a literal result element structure, and can also contain elements that are instructions for creating result tree fragments. When a template is instantiated, each instruction is executed and replaced by the result tree fragment that it creates. Instructions can select and process descendant source node elements. Processing a descendant element creates a result tree fragment by finding the applicable template rule and instantiating its template. The result tree is thus constructed by finding the template rule for the root node, instantiating its template, finding the template rule for a child node, instantiating its template, and recursively repeating these steps until the appropriate template rule has been applied to all nodes of the source tree.

See W3C – XSLT at Section 1. Introduction; Section 5.1. Processing Model.

W3C further describes a structure of a stylesheet. A stylesheet is represented by an xsl:stylesheet element in an XML document, and W3C provides a list of xsl:stylesheet elements including xsl:attribute-set, xsl:variable, and xsl:template, and an example structure of a stylesheet. *See* W3C – XSLT at Section 2.2. Stylesheet Element. A data model used by XSLT for a stylesheet is the same data model used for a source tree and a result tree. *See* W3C – XSLT at Section 3. Data Model.

W3C further describes that a stylesheet includes one or more template rules, where each template rule includes a pattern and a template. A pattern specifies a set of conditions on a source node. A source node that satisfies the conditions matches the pattern, a source node that does not satisfy the conditions does not match the pattern. The template is the template that is applied to the source node that matches the pattern. *See* W3C – XSLT at Section 5.3. Defining Template Rules.

Appellants respectfully submit that W3C fails to disclose or suggest that a stylesheet includes a plurality of cells, where each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell. The Final Office Action cited the "template rules" feature of a stylesheet as allegedly disclosing this limitation. *See* Final Office Action at pages 2-3. Appellants respectfully submit that this position is incorrect. W3C describes that a stylesheet includes a template rule specified with an xsl:template element. The xsl:template element includes a match attribute that identifies a source node or nodes to which the rule applies. *See* W3C – XSLT at Section 5.3. Defining Template Rules. For example, an XML

document might contain the text "This is an <emph>important</emph> point." In the example, the following template rule would match "emph" elements and produce an inline-sequence formatting object with a font-weight property of bold (i.e., produce text where the marked text is bolded, s*ee* W3C – XSLT at *id*:

```
<xsl:template match="emph">
  <fo:inline-sequence font-weight="bold">
    <xsl:apply-templates/>
  </fo:inline-sequence>
</xsl:template>
```

However, Appellants respectfully submit that the match attribute is not a formula specifying an action or computation to perform, where the formula of a first cell may reference a value of the second cell because the match attribute does not reference a value of the second cell. Instead, the match attribute is used to define a pattern that can be applied to a source node in order to determine whether to apply a template to the source node. W3C fails to disclose or suggest that the match attribute references a value of another cell.

The xsl:template element also includes a select attribute that is used to process nodes selected by an expression instead of processing all children of a node that matches the pattern of the template rule. In normal XSLT processing, once a node matches a pattern of a template rule, the stylesheet applies the template to the selected node and all its children. However, the use of a select attribute can be used to modify the normal processing to apply the template to a node set generated by an expression. *See* W3C – XSLT at Section 5.4. Applying Template Rules. For example, the following template rule processes all the children of a node that matches the pattern "author-group", but only if the children are also authors, s*ee* W3C – XSLT at *id*:

```
<xsl:template match="author-group">
  <fo:inline-sequence>
    <xsl:apply-templates select="author"/>
  </fo:inline-sequence>
</xsl:template>
```

Appellants respectfully submit that that the select attribute is not a formula specifying an action or computation to perform, where the formula of a first cell may reference a value of the second cell because the select attribute does not reference a value of the second cell. Instead, the select attribute uses an expression to generate a node set in order to control processing of a selected node's children. W3C fails to disclose or suggest that the select attribute references a value of another cell.

The Final Office Action further took the position that the "xsl:value-of element" of a template discloses a formula of a first cell referencing a value of a second cell. *See* Final Office Action at page 8. Appellants respectfully submit that this position is incorrect as well. W3C describes that within a template, the xsl:value-of element is used to compute generated text, by extracting text from the source tree or by inserting the value of a variable, using an expression that is specified as the value of the select attribute. *See* W3C – XSLT at Section 7.6.1. Generating Text with xsl:value-of. For example, the following template element creates an HTML paragraph from a person element with given-name and family-name attributes, s*ee* W3C – XSLT at *id*:

```
<xsl:template match="person">
  <p>
   <xsl:value-of select="@given-name"/>
   <xsl:text> </xsl:text>
   <xsl:value-of select="@family-name"/>
  </p>
</xsl:template>
```

However, Appellants respectfully submit that that the expressions "@given-name"

and "@family-name" refer to values of a source node, rather than a value of a template element that is distinct from the template element where the expressions are found. Thus, W3C fails to disclose or suggest that the xsl:value-of element of W3C is a cell that includes a formula that references a value of a second cell.

Appellants respectfully submit that W3C also fails to disclose or suggest that a stylesheet includes a plurality of cells, where one of the cells is reserved as an output cell for outputting a result of the processing. The Final Office Action cited the "xsl:output" feature of a stylesheet as allegedly disclosing this limitation. *See* Final Office Action at page 3. Appellants respectfully submit that this position is incorrect. W3C describes a stylesheet element, xsl:output, that allows an XSLT processor to output a result tree formed by the stylesheet. *See* W3C – XSLT at Sec. 16. Output. W3C further describes an XML output method that outputs the result tree as a well-formed XML external general parsed entity, an HTML output method that outputs the result tree as HTML, and a text output method that outputs the result tree as text. *See* W3C – XSLT at Section 16.1. XML Output Method; Section 16.2. HTML Output Method; and Section 16.3. Text Output Method. Appellants respectfully submit that W3C describes that an XSLT processor may output the result tree using the xsl:output element of a stylesheet, but that it is not required to be able to do so. *See* W3C – XSLT at Sec. 16. Output. Thus, W3C fails to disclose or suggest that the xsl:output element of W3C is a cell reserved as an output cell for outputting a result of processing.

Furthemore, Appellants respectfully submit that W3C also fails to disclose or suggest that a stylesheet includes a plurality of cells, where each cell is interlocked with at least one other cell through the formula or attribute of each cell. The Final Office Action

took the position that the "xsl:value-of element" of a template discloses a cell that is interlocked with at least one other cell through the formula or attribute of each cell. *See* Final Office Action at page 4. Appellants respectfully submit that this position is incorrect as well. As previously described, W3C describes that within a template, the xsl:value-of element is used to compute generated text, by extracting text from the source tree or by inserting the value of a variable, using an expression that is specified as the value of the select attribute. *See* W3C – XSLT at Section 7.6.1. Generating Text with xsl:value-of. As previously described, the expression "@given-name" refers to a value of a source node, rather than a value of a template element that is distinct from the template element where the expressions are found. Thus, W3C fails to disclose or suggest that the xsl:value-of element of W3C is a cell that is interlocked with at least one other cell through the formula or attribute of each cell.

In addition, notwithstanding the clear deficiencies of W3C described above (which warrant reversal on their own), the Final Office Action mixes and matches features from each of three discrete components described in W3C to arrive at the claimed "data processing specification comprising a plurality of cells," despite the fact that the Final Office Action has failed to articulate any reasoning with some rational underpinning for doing so. More specifically, independent claim 1 recites a "data processing specification comprising a plurality of cells," and further recites cell-specific limitations which further define a cell. Specifically, independent claim 1 recites that:

- "each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference

a value of a second cell;"

- each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing;" and
- "each cell is interlocked with at least one other cell through the formula or attribute of each cell.

Thus, independent claim 1 makes clear that the aforementioned limitations relate back to the original component of the claim, the "data processing specification." In contrast, W3C clearly indicates that a source tree, a style sheet, and a result tree are three discrete components of an XSLT processing model, where a style sheet is used to transform a source tree into a result tree. See W3C – XSLT at Section 1. Introduction. There is no disclosure or suggestion in W3C of combining any of the three discrete components of the XSLT processing model. Yet, in rejecting the claim, the Final Office Action cites to certain features of the source tree, certain features of the style sheet, and certain features of the result tree, to support its conclusion that claim 1 is obvious, despite the lack of suggestion in W3C of combining any of the three discrete components. *See e.g.,* Final Office Action at pages 2-4. The Final Office Action further fails to articulate any reasoning as to one of ordinary skill in the art would combine separate features of the source tree, style sheet, and result tree described in W3C, to arrive at the claimed "data processing specification" of claim 1.

Accordingly, Appellants respectfully submit that W3C fails to disclose or suggest, "a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is

executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell, wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing … wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in claim 1. Therefore, it is respectfully requested that this rejection be reversed and the claim allowed.

ii). Claim 5

Claim 5 is dependent upon claim 1, and further recites that the first cell has a first attribute referencing a second attribute of said second cell.

Appellants respectfully submit that W3C fails to disclose or suggest, "wherein the first cell has a first attribute referencing a second attribute of said second cell," as recited in claim 5.

The Final Office Action took the position that the "xsl:value-of element" of a template discloses a first cell having a first attribute referencing a second attribute of a second cell. *See* Final Office Action at page 4. Appellants respectfully submit that this position is incorrect. As previously described, W3C describes that within a template, the xsl:value-of element is used to compute generated text, by extracting text from the source tree or by inserting the value of a variable, using an expression that is specified as the value of the select attribute. *See* W3C – XSLT at Section 7.6.1. Generating Text with xsl:value-of. As previously described, the expression refers to a value of a source node, rather than a value of a template element that is distinct from the template element where the expressions are found. Thus, W3C fails to disclose or suggest that the xsl:value-of

element of W3C is a first cell having a first attribute referencing a second attribute of a second cell.

In addition, claim 5 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

iii). Claim 6

Claim 6 is dependent upon claim 1, and further recites that the second cell comprises a reserved mnemonic for providing input to the data processing specified by the data processing specification.

Appellants respectfully submit that W3C fails to disclose or suggest, "wherein said second cell comprises a reserved mnemonic for providing input to the data processing specified by the data processing specification," as recited in claim 6.

The Final Office Action took the position that an attribute value template discloses a reserved mnemonic for providing input to the data processing specified by the data processing specification. *See* Final Office Action at page 5. Applicants respectfully submit that this position is incorrect. W3C merely discloses that in an attribute value template, a value of a src attribute of an img element can be computed from a value of an image-dir variable and the string-value of a href child of a photograph element, and fails to disclose or suggest a reserved mnemonic for providing input. *See* W3C at Section 7.6.2. Attribute Value Templates. In fact, the cited portion of W3C makes no mention of the term "mnemonic." Thus, W3C fails to disclose or suggest that an attribute value template is second cell including a reserved mnemonic for providing input to the data

processing specified by the data processing specification.

In addition, claim 6 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

iv). Claim 7

Claim 7 is dependent upon claim 1, and further recites that the first cell is a reserved output cell specification specifying output of the data processing specified by the data processing specification.

Appellants respectfully submit that W3C fails to disclose or suggest, "wherein said first cell is a reserved output cell specification specifying output of the data processing specified by the data processing specification," as recited in claim 7.

The Final Office Action cited the "xsl:output" feature of a stylesheet as allegedly disclosing a reserved output cell specification specifying output of the data processing specified by the data processing specification. *See* Final Office Action at page 3. Appellants respectfully submit that this position is incorrect. As previously described, W3C describes a stylesheet element, xsl:output, that allows an XSLT processor to output a result tree formed by the stylesheet. *See* W3C – XSLT at Sec. 16. Output. W3C further describes an XML output method that outputs the result tree as a well-formed XML external general parsed entity, a HTML output method that outputs the result tree as HTML, and a text output method that outputs the result tree as text. *See* W3C – XSLT at Section 16.1. XML Output Method; Section 16.2. HTML Output Method; and Section 16.3. Text Output Method. Appellants respectfully submit that W3C describes that an XSLT

processor may output the result tree using the xsl:output element of a stylesheet, but that it is not required to be able to do so. *See* W3C – XSLT at Sec. 16. Output. Thus, W3C fails to disclose or suggest that the xsl:output element of W3C is a reserved output cell specification specifying output of the data processing specified by the data processing specification.

In addition, claim 7 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

v). Claim 8

Claim 8 is dependent upon claim 1, and further recites that the second cell comprises a conditionally executed formula. Claim 8 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

vi). Claim 9

Claim 9 is dependent upon claim 1, and further recites that the data processing specification further includes one or more global attributes specifying one or more global processing characteristics for the specified data processing.

Appellants respectfully submit that W3C fails to disclose or suggest, "wherein said data processing specification further includes one or more global attributes specifying one or more global processing characteristics for the specified data processing," as recited in claim 9.

The Final Office Action cited the "xsl:stylesheet version" and "xmlns:xsl" features of a stylesheet as allegedly disclosing one or more global attributes specifying one or more global processing characteristics for a specified data processing. *See* Final Office Action at page 5. Appellants respectfully submit that this position is incorrect. W3C describes that "xsl:stylesheet version" is merely a version attribute that indicates the version of XSLT that the stylesheet requires. *See* W3C – XSLT at Section 2.2. Stylesheet Element. W3C fails to disclose or suggest that the version attribute is used by any of the stylesheet elements to process a source tree and convert it to a result tree. Furthermore, W3C describes that "xmlns:xsl" is merely a XSLT namespace attribute which is used to identify the namespace used by the stylesheet and to recognize elements and attributes within the stylesheet. *See* W3C – XSLT at Section 2.1. XSLT Namespace. W3C fails to disclose or suggest that the stylesheet uses the namespace to process a source tree and convert it to a result tree. Thus, W3C fails to disclose or suggest that the xsl:stylesheet version and xmlns:xsl elements of W3C are global attributes specifying one or more global processing characteristics for a specified data processing.

In addition, claim 9 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

vii). Claim 10

Claim 10 is dependent upon claim 9, and further recites that the one or more global attributes include a global attribute specifying a format for providing the specified data processing with an HTTP request.

Appellants respectfully submit that W3C fails to disclose or suggest, "wherein said one or more global attributes include a global attribute specifying a format for providing the specified data processing with an HTTP request," as recited in claim 10.

The Final Office Action cited the "xmlns:xsl" feature of a stylesheet as allegedly disclosing a global attribute specifying a format for providing the specified data processing with an HTTP request. *See* Final Office Action at page 5. Appellants respectfully submit that this position is incorrect. As previously described, W3C describes that "xmlns:xsl" is merely a XSLT namespace attribute which is used to identify the namespace used by the stylesheet and to recognize elements and attributes within the stylesheet. *See* W3C – XSLT at Section 2.1. XSLT Namespace. W3C fails to disclose or suggest that the stylesheet uses the namespace to specify a format for providing a specified data processing with an HTTP request. Thus, W3C fails to disclose or suggest that the xmlns:xsl element of W3C is a global attribute specifying a format for providing the specified data processing with an HTTP request.

In addition, claim 10 is patentable for the same reasons that claim 1 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

viii). Claim 11

Claim 11 recites an apparatus. The apparatus includes at least one storage unit having stored thereon programming instructions that are configured to be executed by a computer processor and designed to receive as input computer code a data processing specification comprising a plurality of cells, where each cell comprises a formula

specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, where the formula of a first cell may reference a value of a second cell. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. The programming instructions are further designed to parse the specification to determine an interdependency of the plurality of cells and generate and store a directed graph of the interdependency as an execution flow. The programming instructions are further designed to execute the computer code of the specification in accordance with the execution flow, where the executing comprises evaluating the formula of each cell in the execution flow and generating an output result. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. The apparatus further includes at least one processor coupled to said at least one storage unit to execute said programming instructions.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell, wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing ... wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in claim 11 based on reasoning similar to the reasoning discussed in Section VII, A, i.

Therefore, it is respectfully requested that this rejection be reversed and the claim allowed.

ix). Claim 15

Claim 15 is dependent upon claim 11, and further recites that the programming instructions are designed to support the first cell having a first attribute referencing a second attribute of said second cell.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "the first cell having a first attribute referencing a second attribute of said second cell," as recited in claim 15 based on reasoning similar to the reasoning discussed in Section VII, A, ii.

In addition, claim 15 is patentable for the same reasons that claim 11 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

x). Claim 16

Claim 16 is dependent upon claim 11, and further recites that the programming instructions are designed to support said second cell having a reserved mnemonic for facilitating provision of input to the data processing specified by the data processing specification.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "said second cell having a reserved mnemonic for facilitating provision of input to the data processing specified by the data processing

specification," as recited in claim 16 based on reasoning similar to the reasoning discussed in Section VII, A, iii.

In addition, claim 16 is patentable for the same reasons that claim 11 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

xi). Claim 17

Claim 17 is dependent upon claim 11, and further recites that the programming instructions are designed to support said first cell being a reserved output cell specification specifying output of the data processing specified by the data processing specification.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "said first cell being a reserved output cell specification specifying output of the data processing specified by the data processing specification," as recited in claim 17 based on reasoning similar to the reasoning discussed in Section VII, A, iv.

In addition, claim 17 is patentable for the same reasons that claim 11 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

xii). Claim 18

Claim 18 is dependent upon claim 11, and further recites that the programming instructions are designed to support said second cell having a conditionally executed

formula. Claim 18 is patentable for the same reasons that claim 11 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

xiii). Claim 19

Claim 19 is dependent upon claim 11, and further recites that the programming instructions are designed to support said data processing specification having one or more global attributes specifying one or more global processing characteristics for the specified data processing.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "wherein said data processing specification further includes one or more global attributes specifying one or more global processing characteristics for the specified data processing," as recited in claim 19 based on reasoning similar to the reasoning discussed in Section VII, A, vi.

In addition, claim 19 is patentable for the same reasons that claim 11 is patentable. Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

xiv). Claim 20

Claim 20 is dependent upon claim 19, and further recites that the programming instructions are designed to support one of said one or more global attributes being a global attribute specifying a format for providing the specified data processing with an HTTP request. Claim 20 is patentable for the same reasons that claim 11 is patentable.

Accordingly, it is respectfully requested that this rejection be reversed and the claim allowed.

xv). Claim 21

Claim 21 recites a computer with a memory having stored thereon instructions that when executed cause to the computer to implement data processing. The computer includes means for receiving a data processing specification comprising a plurality of cells. Each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells. The formula of a first cell may reference a value of a second cell. Each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing. The computer further includes means for parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow. The computer further includes means for executing the specification in accordance with the execution flow. The executing comprises evaluating the formula of each cell in the execution flow and generating an output result. Each cell is interlocked with at least one other cell through the formula or attribute of each cell. Each cell is interlocked with at least one other cell through the formula or attribute of each cell.

While each of the claims have their own scope, Appellants respectfully submit that W3C fails to disclose or suggest, at least, "a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing

other cells, wherein the formula of a first cell may reference a value of a second cell, wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing ... wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell," as recited in claim 21 based on reasoning similar to the reasoning discussed in Section VII, A, i.

Therefore, it is respectfully requested that this rejection be reversed and the claim allowed.

For all of the above noted reasons, it is strongly contended that certain clear differences exist between the present invention as claimed in claims 1, 5-11, 15-21 and the prior art relied upon by the Examiner. It is further contended that these differences are more than sufficient that the present invention would not have been obvious to a person having ordinary skill in the art at the time the invention was made.

This final rejection being in error, therefore, it is respectfully requested that this honorable Board of Patent Appeals and Interferences reverse the Examiner's decision in this case and indicate the allowability of application claims 1, 5-11, and 15-21.

In the event that this paper is not being timely filed, the applicants respectfully petition for an appropriate extension of time. Any fees for such an extension together with any additional fees which may be due with respect to this paper may be charged to Counsel's Deposit Account 50-2222.

Respectfully submitted,

SQUIRE, SANDERS & DEMPSEY LLP

/Keith M. Mullervy/
Keith M. Mullervy
Attorney for Applicant(s)
Registration No. 62,382

Atty. Docket No.: 104402.00119

8000 Towers Crescent Drive, 14th Floor
Vienna, VA 22182-6212
Tel: (703) 720-7843
Fax (703) 720-7802

KMM:jf

Encls: Appendix 1 - Claims on Appeal
Appendix 2 - Evidence
Appendix 3 - Related Proceedings

APPENDIX 1

CLAIMS ON APPEAL

1. (Previously Presented) A computer-implemented method of cell-based data processing that facilitates the execution of computer programming code by a computer system, the method comprising:

receiving as input computer code a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell;

wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing;

parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow; and

executing the computer code of the specification in accordance with the execution flow, wherein the executing comprises evaluating the formula of each cell in the execution flow and generating an output result;

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell.

2. - 4. (Cancelled).

5. (Previously Presented) The method of claim 1, wherein the first cell has a first attribute referencing a second attribute of said second cell.

6. (Previously Presented) The method of claim 1, wherein said second cell comprises a reserved mnemonic for providing input to the data processing specified by the data processing specification.

7. (Previously Presented) The method of claim 1, wherein said first cell is a reserved output cell specification specifying output of the data processing specified by the data processing specification.

8. (Previously Presented) The method of claim 1, wherein said second cell comprises a conditionally executed formula.

9. (Original) The method of claim 1, wherein said data processing specification further includes one or more global attributes specifying one or more global processing characteristics for the specified data processing.

10. (Original) The method of claim 9, wherein said one or more global attributes include a global attribute specifying a format for providing the specified data processing with an HTTP request.

11. (Previously Presented) An apparatus comprising:

at least one storage unit having stored thereon programming instructions that are configured to be executed by a computer processor and designed to:

receive as input computer code a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell;

wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing;

parse the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow; and

execute the computer code of the specification in accordance with the execution flow, wherein the executing comprises evaluating the formula of each cell in the execution flow and generating an output result;

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell; and

at least one processor coupled to said at least one storage unit to execute said programming instructions.

12. - 14. (Cancelled).

15. (Previously Presented) The apparatus of claim 11, wherein said programming instructions are designed to support the first cell having a first attribute referencing a second attribute of said second cell.

16. (Previously Presented) The apparatus of claim 11, wherein said programming instructions are designed to support said second cell having a reserved mnemonic for facilitating provision of input to the data processing specified by the data processing specification.

17. (Previously Presented) The apparatus of claim 11, wherein said programming instructions are designed to support said first cell being a reserved output cell specification specifying output of the data processing specified by the data processing specification.

18. (Previously Presented) The apparatus of claim 11, wherein said programming instructions are designed to support said second cell having a conditionally executed formula.

19. (Previously Presented) The apparatus of claim 11, wherein said programming instructions are designed to support said data processing specification having one or more global attributes specifying one or more global processing characteristics for the specified data processing.

20. (Original) The apparatus of claim 19, wherein said programming instructions are designed to support one of said one or more global attributes being a global attribute specifying a format for providing the specified data processing with an HTTP request.

21. (Previously Presented) A computer with a memory having stored thereon instructions that when executed cause to the computer to implement data processing comprising:

means for receiving a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell;

wherein each cell is delineated by a beginning and ending tag, and one of the cells is reserved as an output cell for outputting a result of the processing;

means for parsing the specification to determine an interdependency of the plurality of cells and generating and storing a directed graph of the interdependency as an execution flow; and

means for executing the specification in accordance with the execution flow, wherein the executing comprises evaluating the formula of each cell in the execution flow and generating an output result;

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell.

22-25. (Cancelled).

APPENDIX 2

EVIDENCE APPENDIX

No evidence under section 37 C.F.R. 1.130, 1.131, or 1.132 has been entered or will be relied upon by Appellants in this appeal.

APPENDIX 3

RELATED PROCEEDINGS APPENDIX

No decisions of the Board or of any court have been identified under 37 C.F.R. §41.37(c)(1)(ii).